JhoveIntegration

Jhove Integration with DSpace

Update - June 28, 2006

The Jhove integration project has changed shape since the last time I posted code. After looking more closely at the capabilities of Jhove and at our plans for other tools, we decided (unfortunately) to scrap my original code in ItemImport. Please see below for more info. As always, feedback is welcome.

What Jhove Does

Jhove is a tool created by JSTOR and the Harvard University Library that, when passed a file (or set of files), will identify the file format, determine whether it is a well-formed/valid instance of that format, and will also extract technical metadata from the file. For a more detailed description of Jhove, see the Jhove website.

Integration with DSpace

At first we hoped to use Jhove for file format identification, metadata extraction, and format validation for all bitstreams upon ingest. However, for a number of reasons, including questions about the reliability of Jhove's format identification abilities, uncertainty about how we'll use the extracted metadata, and probable changes in the tool landscape, we've changed our original plan a bit. We've divided the project into two parts~~ a command-line tool and integration into DSpace workflow ~~and we've narrowed down our use of Jhove:

For ingest, use Jhove for file format validation only

Since Jhove's format identification functionality seems somewhat unreliable, for now we are sticking with DSpace's identification based on file extensions, and will use Jhove only for format validation on ingest. Technical metadata extraction will be available only via a command-line tool (see below). Hopefully there will be another tool in the not-too-distant future that will provide more reliable format identification (either Jhove2 or the UK National Archive's DROID tool).

Integrate Jhove into the DSpace Workflow code, so that the code can be used from any of the three ingest methods (import, Web UI, LNI)

The workflow code is already called/calleable from all three ingest methods, so it provides a centralized place for making calls to Jhove. Unfortunately, the existing workflow steps are all hard-coded, with hard-coded fields in the database. Ideally we would re-write the workflow code to create a series of configurable steps, or even integrate a third-party workflow engine. However, we don't have the resources for that on this project, so I propose (following Richard Rodgers' suggestion) a "shallow" integration that won't involve a lot of code modifications, but that will provide a starting point for using Jhove or other tools on ingest.

Workflow Modifications for Integrating Jhove

1. Add a hard-coded (ugh) "Pre-ingest" workflow step.
2. Unlike other workflow steps, this step will be turned on or off via a line in the dspace.cfg file. The on/off toggle will apply to **all** collections; it will not be available to individual collections. There will be no way to add/delete/edit this step via the GUI.
3. The e-person group associated with this step will always be the admin group associated with a particular collection.
4. If Jhove validates a file, the workflow step will be skipped. If Jhove determines that a file is invalid, then an e-mail will be sent to members of the admin group. When they log in to DSpace, they will be presented with an item in their task pool, and they will have the choice of either accepting or rejecting the item. If accepted, the item will then proceed through all the regular workflow steps, if they're in place.
5. This pre-ingest step could potentially be used for other tools, such as virus-checking, file identification, and perhaps even file migration. The plugin Manager could be used to determine which tools should be used.

Create a command-line tool for running Jhove on an item/collection/community on an as-needed basis

A command-line tool side-steps the thorny issue of how to save and access the extracted metadata, since in theory it could be extracted at any time. And since we're not sure yet how we'd use the metadata extracted by Jhove, it doesn't make sense to spend lots of time right now working on a way to save and access it.

I've completed an alpha version of the command-line tool. The code can be accessed at http://libaxis1.mit.edu/viewcvs/sandbox/TechMDExtractor.